# Research Trends in Simulation of Networked Control Systems

Manisha Patel

*Department of Instrumentation & Control Engineering*
*L.D. College of Engineering*
*Ahmedabad*

Prof. V.P.Patel

*Head of department(I.C)*
*Department of Instrumentation & Control Engineering*
*L.D. College of Engineering*
*Ahmedabad*

***Abstract*** **– Networked Control Systems (NCSs) are one type of distributed control systems where sensors, actuators controllers are interconnected by communication networks. In this paper the NCS and its simulations are discussed. The beginning of this paper discusses the NCS and control and communication challenges in NCS. The next part of this paper discusses the simulation of NCS using different simulators and control methodologies.**

***Keywords*** – Networked Control System (NCS), Matlab/Simulink, ns-2

## I. INTRODUCTION

When a traditional feedback control system is closed via a communication channel (such as network); which may be shared with other nodes outside the control system, then the control system is classified as a Networked Control System (NCS) [1]. Digital computers became powerful tools in control system design and microprocessors added a new dimension to the capability of control systems. As the expanding needs of industrial applications pushed the limit of point to point control, it became obvious that NCS was the solution to achieve remote control operations.

NCS has defining feature is that information is exchanged among control system components (sensor, controller, actuator etc.) using a shared network (Figure 1).

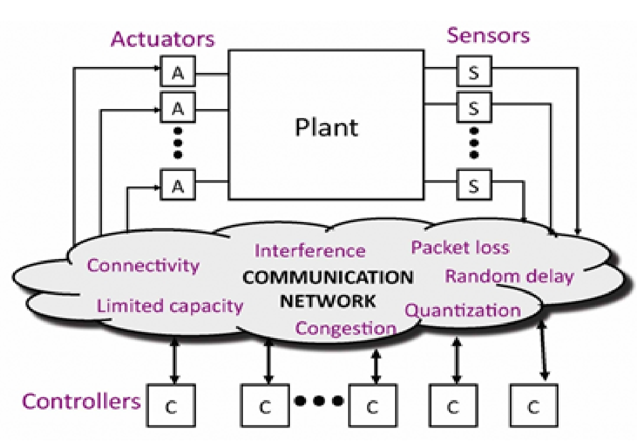


Figure 1.A typical structure of networked control system.

In broader terms, NCS research is categorized into two parts:

(1) *Control of Network*: Study and research on communications and networks to make them suitable for real time NCS, e.g. routing control, congestion reduction, efficient data communication, networking protocol etc.

(2) *Control over Network*: Mainly it deals with control strategies and control systems design over the network to minimize the effect of adverse network parameters on NCS performance such as network delay.

Again the "*Control over Network"* there are two major types of control systems that utilize communication networks: (1) Shared network control systems and (2) Remote control systems.

As NCS become increasingly complex, it become more challenging to formally analyze their performance, stability, safety and security properties [7]. So there is a pressing need to evaluate both the control and network components of NCS together for a rapidly growing number of applications, such as unmanned aerial vehicles (UAVs) and industrial control systems. Simulation is a powerful technique for evaluation and can be used at various design stages, but it requires the support of appropriate tools during both the design time and run time stages in order for the process to be efficient and less prone to errors.

This paper is organized as follows. Section II presents control and communication challenges. Section III presents study of different simulators used for simulating NCS. Section IV presents control methodologies. Section V presents conclusion.

## II. CONTROL AND COMMUNICATION CHALLENGES IN NCS

When sensors and actuators communicate with a remote controller over a multipurpose network, improved techniques are needed for closed loop stability and controller synthesis [2].

Each of the NCS structure has many challenges to maintain the Quality of Service (QoS) and Quality of Control (QoC). In the network, QoS is the idea that transmission rates, error rates and other characteristics can be measured, improved and to some extent guaranteed in advance [3]. The QoS can be degraded due to congestion and interference.



ISSN: 2278 – 7798

Implementation issues in both hardware and software are at the center of successful deployment of NCS. Data integrity and security are also very important and may lead to special consideration in control system design even at early stages [4].

From the beginning of the NCS systems some important areas are always focus in field of research. Broadly speaking these are 1) the assurance of control performance in the face of communication constraints between network nodes; 2) the effect of latency, network overhead, noise, delay and packet dropouts; 3) Consensus, coverage and optimized cooperation in systems of multiple mobile agents; and 4) information patterns for decentralized control of networks of mobile agents. Specifically real time requirements suggest new approaches to the design of the network protocols as solutions to optimization problems.

Network induced delays and packet dropouts are two major causes for the NCSs performance deterioration and potential NCSs instability [5]. The network induced delay in NCSs occurs when sensors, actuators and controllers exchanges data across the network. Depending on the medium access control (MAC) protocol of the control network, network-induced delay can be constant, time varying or even random. MAC protocol generally falls into two categories: *random access* and *scheduling* [6]. Carrier sense multiple access (CSMA) is most often used in random access networks, whereas token passing and time division multiple access (TDMA) are commonly employed in scheduling networks. Control networks using CSMA protocols include Device Net and Ethernet. Figure 2 illustrates various possible situations for this type of network.

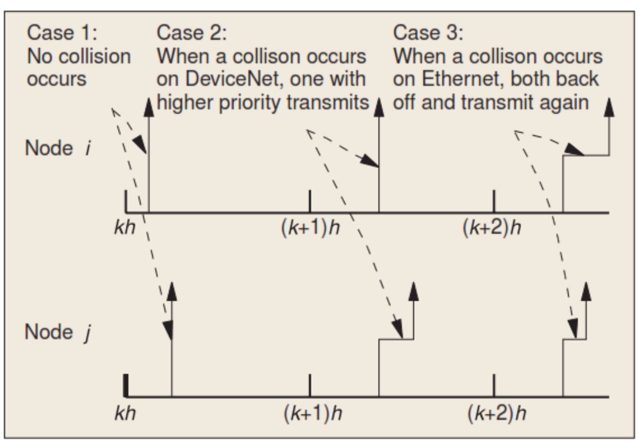


Figure 2. Timing diagram for two nodes ona random access network

The figure depicts two nodes continually transmitting messages (with respect to a fixed time line). A node on a CSMA network monitors the network before each transmission. When the network is idle, it begins transmisssion immediately as shown in case 1 of figure 2. Otherwise it waits until the network is not busy. When two or more nodes try to transmit simultaneously, a collision occurs. The way to resolve the collision is protocol dependent. Device Net, which is a controller area network (CAN), uses CSMA with a bitwise arbitration protocol. Since CAN messages are prioritized, the message with the hifhest prioroty is transmitted without interruption when a collision occurs, and transmision of the lower priority message is terminated and will be retried when the network is ideal, as shown in case 2 of figure 2. Ethernet employs a CSMA with collision detection (CSMA/CD) protocol. When there is a collision, all of the affected nodes will back off, waits a random time and retransmit, as shown in case 3 of figure 2. Packets on these type of networks are affected by random delays and the worst-case transmission time of packets is unbounded.

When using an NCS, one must consider not only network induced delay, but also data packet dropout. Networks can be viewed as unreliable data transimission paths, where packet collision and network node failure occasionally occurs. When there is a packet collision, instead of repeated retransmission attempts, it might be advantageous to drop the old packet and transmit a new one. Thus it is valuable to analyze the rate at which the data should be transmitted to achieve the desired performance(stability). An NCS with data packet dropout can be modeled as an asynchronous dynamical system (ADS) with rate constrints on events.

Some important control methodologies, such as stochastic optimal control, robust control, predictive control and state feedback control have been proposed to address the problems of network-induced delays or packet dropouts.

## III. Simulation of NCS using different simulators

Some general simulators such as ns-2 and OMNeT++ which are discrete event simulators are often used for the simulation of NCS [8]. These simulators lack the ability to simulate continuous time dynamics as well as the flexibility and ease of modeling complex dynamical systems.

Rather than model NCS using general network simulators, some approaches co-simulate NCS using powerful simulators for dynamical systems and then incorporate network abstractions. A typical example of this approach is Truetime. Truetime extends Matlab/Simulink with platform related modeling concepts (i.e., networks, clocks, schedulers) and supports simulation of networked and embedded control systems with implementation effects [9]. Modeling the network dynamics in Truetime is highly abstracted, and therefore, limits the level of details of the network layers that can be modeled and implemented. For example, in TrueTime it is not straightforward to specify routing protocols or network queuing management schemes.

Recently there have been advances in various approaches that integrate or couple multiple simulators together in order to effectively simulate NCS. In [10], a tool chain called PiccSIM was developed, which allows the integration of Matlab/Simulink models with ns-2. PiccSIM provides a graphical user interface for the design of networked control systems and the automatic code generation of ns-2 scripts and



ISSN: 2278 – 7798

Matlab/Simulink models. Also in [11], a special simulator coupling concept implemented in C/C++ is used to integrate the simulators ModelSim, Matlab/Simulink and ns-2. Although these two approaches address integration of simulators, the integration is not based on a standard framework, so it is not possible to reason about the accuracy or extensibility of the approaches.

The HLA is a standard for simulation interoperability originally developed by the U.S. Department of Defence [8]. IEEE now oversees the on-going evolution of the architecture as the HLA Standard 1516. The HLA consist of three components: (1) Federation Rules that define the basic principles underlying the HLA and describe the simulation and federate responsibilities; (2) Interface Specification that defines the interface to the Run-Time Infrastructure (RTI) services that provide the means for simulators to coordinate the execution and exchange of information; (3) Object Model Template (OMT) that provides a standard format for describing information of common interest to the federates. An HLA simulation typically consists of a collection of simulators each simulator is called federates. Communications between different federates is managed by the Run-Time Infrastructure (RTI). The RTI provides a set of services such as time management, data distribution, and ownership management. Since HLA is an accepted standard, a number of RTI implementations are available [13].

In order to develop a realistic and accurate simulation of NCS, we need a modeling and simulation environment that can integrate existing tools for the accurate simulation of the control dynamics as well as the networking system of a NCS. An integrated modeling and simulation tool for NCS, called the Networked Control Systems Wind Tunnel (NCSWT), which combines the network simulation capabilities of ns-2 with the control design and simulation capabilities of Matlab/Simulink [14]. The architecture of NCSWT which mainly consist of two part design time and run time. The design time models constructs from given DSMLS generate software components, interfacing glue code and configuration files for the NCS which is deployed and executed at run time. The Three DSMLs are: the NCSWT model integration language (NCSWTMIL), the Control design modeling language (CDML), the Network design modeling language (NDML). Here CDML does not re implement all of the Matlab/Simulink syntax and blocks, it provides an interface for generating and building specified control design models based on a user defined library of Simulink blocks and inputs for a NCS. NDML does not re implement the ns-2 syntax and simulation semantics. It essentially provides an interface language for defining the network interaction between the ns-2 simulation and the Matlab simulation.

The modeling of the NCS is performed in GME using the three DSMLs. The modeling and code generation is performed on a computer running a Windows operating system. The second step involves the deployment of the generated code and models and the execution of the simulation. The simulation is executed on a computer running a Linux operating system. To build the NCSWT tool required software are: Matlab/Simulink, ns-2, Portico 1.0.2, Generic Modeling Environment (GME), Universal Data Model (UDM), Microsoft visual studio 2008 or later, Eclipse.

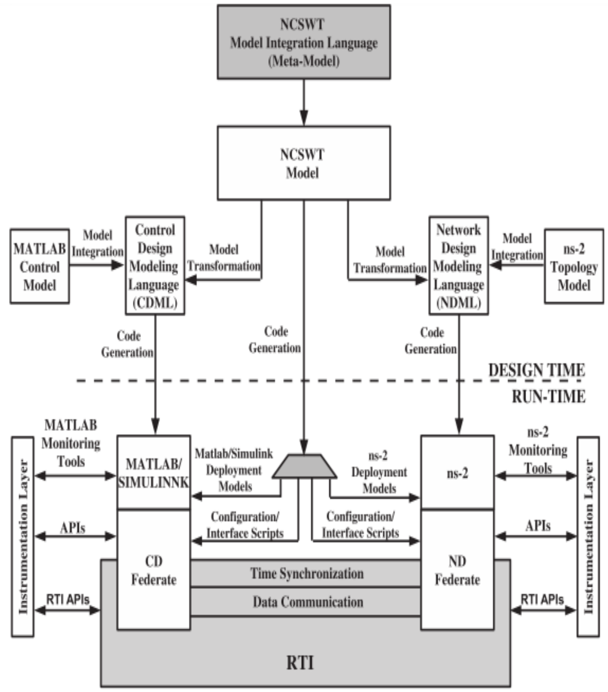


Figure 3. Overview of NCSWT

NCSWT allows a user to rapidly reconfigure their experimental setup for the simulation of NCS. For example, in order to evaluate the impact of network effects on a NCS, the configurations in the NDML is modified to the desired network configuration setup and then updated code is generated from the network model without modifying the control models. Similarly, if the dynamics of the control system is changed, the models in the CDML are modified to reflect the changes while maintaining the same network configurations, and then updated control models are generated.

## IV. CONTROL METHODOLOGIES IN NCS

Due to network delay concerns, the methodologies to control an NCS have to maintain the stability of the system in addition to controlling and maintaining the system performance as much as possible [15]. Some methodologies are: 1) *Augmented deterministic discrete-time model methodology*- Halevi and Ray (1988) proposed a methodology named as the augmented deterministic discrete time model methodology to control a linear plant over periodic delay network. The structure of the augmented discrete-time model is straightforward and can be modified to support non-



ISSN: 2278 – 7798

identical sampling periods of a sensor and a controller. 2) *Queuing methodology-* Queuing mechanisms can be used to reshape random network delays on an NCS to deterministic delays such that the NCS becomes time-invariant. These methodologies have been developed by utilizing some deterministic or probabilistic information of an NCS for the control algorithm formulation. An early queuing methodology was developed by Luck and Ray (1990, 1994) denoted here as the deterministic predictor-based delay compensation methodology. This methodology uses an observer to estimate the plant states and a predictor to compute the predictive control based on past output measurements. 3) *Optimal stochastic control methodology-* Nilsson (1998) proposed the optimal stochastic control methodology to control an NCS on random delay networks. The optimal stochastic control methodology treats the effects of random network delays in an NCS as a Linear–Quadratic–Gaussian (LQG) problem other than the assumptions mentioned earlier, this methodology assumes that $\tau < T$. 4) *Perturbation methodology-* Walsh, Beldiman, Ye, and Bushnell (1999, 1999)used non-linear and perturbation theory to formulate network delay effects in an NCS as the vanishing perturbation of a continuous-time system under the assumption that there is no observation noise. It can be applied on an NCS on periodic delay networks and random delay networks at the sensor-to controller transmission. However, these networks are restricted to be priority-based networks, which can assign different priorities to data transmissions. These priorities can be managed by priority scheduling algorithms proposed in Walsh, Beldiman, and Bushnell (1999). In addition, this methodology requires a very small sampling time so that an NCS can be approximated as a continuous-time system. A control loop in the perturbation methodology consists of a nonlinear controller and a nonlinear plant, but the analysis and derivations used can be similarly applied to linear systems.6) *Sampling time scheduling methodology-* Hong (1995) developed the sampling time scheduling methodology to appropriately select a sampling period for an NCS such that network delays do not significantly affect the control system performance, and the NCS remains stable. This methodology is originally used for multiple NCS on a periodic delay network, in which all connections of every NCS on the network are known in advance. This methodology requires $\tau < T$, and is applicable to only a single-dimensional NCS. 6) *Robust control methodology-* Goktas (2000) designed a networked controller in the frequency domain using robust control theory. A major advantage of this methodology is that it does not require a priori information about the probability distributions of network delays. In the robust control methodology, the network delays $\tau^{ca}$ and $\tau^{sc}$ are modeled as simultaneous multiplicative perturbation. Both delays $\tau^{ca}$ and $\tau^{sc}$ are also assumed to be bounded and able to be approximated by the fluid flow model. 7) *Fuzzy logic modulation methodology-* Almutairi et al., (2001) proposed the fuzzy logic modulation methodology for an NCS with a linear plant and a modulated PI controller to compensate the network delay effects based on fuzzy logic In this methodology, the PI controller gains are externally updated at the controller output with respect to the system output error caused by network delays. Thus, the PI controller needs not to be redesigned, modified, or interrupted for use on a network environment. 8) *Event-based methodology-* Tarn and Xi (1998) introduced the event-based methodology for networked control of a robotic manipulator over the Internet. Instead of using time, this methodology uses a system motion as the reference of the system. 9) *End-user control adaptation methodology-* Tipsuwan and Chow (2001) proposed the end-user control adaptation methodology. The main concept of end-user control adaptation is to adapt controller parameters (e.g., controller gains) with respect to the current network traffic condition or the current given network Quality-of-Service (QoS). In this methodology, the controller and the remote system are assumed to be able to measure network traffic conditions.

Predictive control algorithm is also proposed to compensate the network delay and achieve desired control performance of NCS [16]. By applying the adaptive Smith predictor with an online estimator for the delay time, the significantly induced time delay effect on the NCS has been successfully reduced [17].

## V. CONCLUSION

Integrating computer networks into control systems to replace traditional point-to-point wiring has enormous advantages but on the other hand NCSs bring several issues like network-induced delays or packet dropouts. We need to know the behavior and characteristics of NCSs for designing better and robust control systems. It is possible by using the NCSWT tool which integrates Matlab/Simulink and ns-2 for modeling and simulation of NCS using the High Level Architecture (HLA) standard. HLA guarantees accurate time synchronization and data communication.

## REFERENCES


[1] R. Gupta, M.-Y. Chow, Networked control system: overview and research trends, IEEE Transactions on Industrial Electronics 57 (7) (2010) 2527–2535.

[2] J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems, Proceedings of the IEEE 95 (1) (2007) 138–162. M. King and B. Zhu, "Gaming strategies," in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.

[3] Soucek S, Sauter T, Quality of service concerns in IP based control systems, IEEE transaction on Industrial Electronics, vol.51, issue:6, pp. 1249-1258, December 2004.

[4] J. Baillieul, P. Antsaklis, Control and communication challenges in networked control systems, Proceedings of the IEEE 95 (1) (2007) 9–28.

[5] Hongbo Li, Zengai Sun, Bacong Chen, Huaping Liu and Fuchun Sun, Intelligent scheduling control of networked control system with networked induced delay and packet dropout. International Journal of control, Automation and system, vol.6, no.6, pp. 915-927, December 2008.

[6] Weizhang, Michael S. Branicky and Stephen M. Phillips, Stability of Networked control systems. IEEE control system magazine 0272-1708/01 2001 IEEE.




ISSN: 2278 – 7798

[7] Emeka Eyisi, Jai Bai, Derek Riley, Jiannian Weng, Wei Yan, Yuan Xue, Xenofon Koutsoukos, Janos Sztipanovits, NCSWT: An integrated Modeling and Simulation tool for Networked Control systems, HSCC 12 April 17-19, 2012, Beijing, China.

[8] D. Riley, E. Eyisi, J. Bai, Y. Xue, X. Koutsoukos, J. Sztipanovits, Networked control system wind tunnel (NCSWT) – an evaluation tool for networked multi-agent systems, in: 4th Int. ICST Conf. on Simulation Tools and Techniques (SIMUTools), 2011.

[9] A.cervin, M.ohlin and D.Henori Ksson, Simulation of networked control systems using true time, In proc 3rd International workshop on networked control systems, tolerant to faults,2007.

[10] T. Kohtamaki, M. Pohjola, J. Brand, L. Eriksson, Piccsim toolchain – design, simulation, and automatic implementation of wireless networked control systems, in: IEEE Conference on Networking, Sensing, and Control, 2009, pp. 49–54.

[11] U. Hatnik, S. Altmann, Using modelsim, matlab/simulink and ns for simulation of distributed systems, International Conference on Parallel Computing in Electrical Engineering 0 (2004) 114–119.

[12] HLA. IEEE standard for modeling and simulation (mamp; s) high level architecture (hla) - framework and rules. IEEE Std. 1516-2000, pages i – 22, 2000.

[13] F. Kuhl, R. Weatherly, and J. Dahmann. Creating Computer Simulation Systems: An Introduction to the High Level Architecture. Prentice Hall, 1999.

[14] Emeka Eyisi, Jai Bai, Derek Riley, Jiannian Weng, Wei Yan, Yuan Xue, Xenofon Koutsoukos, Janos Sztipanovits, NCSWT: An Integrated modeling and simulation tool for networked control systems, Simulation Modeling Practice and theory 27 (2012) 90-111 Elsevier.

[15] Y. Tipsuwan, M.-Y. Chow, Control methodologies in networked control systems, Control Engineering Practice 11 (10) (2003) 1099–1111.

[16] G.P. Liu and D. Ress and S.C. Chai, Design and Implementation of Networked predictive control systems 0-7803- 8823-7 / 05 2005 IEEE

[17] Chien-Liang Lai and Pau Lo Hsu, Design the Remote Control System with the Time Delay Estimator and the Adaptive Smith Predictor, IEEE Transaction on Industrial Informatics Vol. 6 No.1 February -2010.




ISSN: 2278 – 7798